

AMENDMENT TO THE CLAIMS

1. (Currently Amended) A computer-implemented method for reducing coding errors prior to runtime in the context of a managed code execution environment, comprising:

providing a developer with access to a plurality of managed code resources through a graphical user interface, the graphical user interface having a coding area, a tab, a file selection area, and a configuration area, the coding area receiving code input from the developer, the tab indicating a designation of a file that the developer is authoring, the file selection area enabling the developer to move between different files and projects, the configuration area enabling the developer to selectively activate or deactivate various programming tools, one of the various programming tools being a StringRes tool;

utilizing a computer processor that is a functional component of the computer to verify that a resource identifier input by the developer corresponds to one of the plurality of managed code resources by:

receiving from the developer a precursor to the resource identifier in the coding area;

following the receipt of the precursor, receiving an activation key from the developer in the coding area;

initiating, based at least in part on receipt of the activation key and the precursor, a response from the StringRes tool, the response including a drop-down menu that is located proximate the precursor in the coding area, the drop-down menu containing resource information that is valid for the precursor, the information arranged in a hierarchical fashion, the information including key names that correspond to elements;

the developer navigating the drop-down menu and pausing on one of the key names;

initiating, based at least in part on the pause, an appearance of a corresponding pop-up box, the pop-up box located proximate the

drop-down menu in the coding area, the drop-down menu being located between the precursor and the pop-up box, the drop-down menu including string and value information related to the one of the key names;
receiving a selection from the developer of a second one of the key names;
and
providing the developer with a collection of resource identifiers; and
receiving said resource identifier input from the developer in the form of a selection from the collection of resource identifiers; and
automatically inserting, based at least in part on the selection, the one of the plurality of managed code resources into a programming code,
the one of the plurality of manage code resources corresponding to the second one of the key names, the programming code located in the coding area.

2-3. (Cancelled)

4. (Currently Amended) The method of claim 12,4, ~~wherein providing a collection of resource identifiers comprises providing a collection of resource identifiers in response to an input by the developer of an activation key, wherein the activation key is selected from the group consisting of a period, a space bar, and a left parenthesis.~~

5-11. (Cancelled)

12. (Currently Amended) The method of claim 1, further comprising:
receiving from the developer an addition to the plurality of managed code resources, wherein the addition configures the managed code execution environment to accept a new resource input, wherein receiving the addition comprises:
bringing up a menu of functional options by choosing a project in the file selection area;

selecting an add new item function from the menu;
bringing up a dialog box in response to the add new item function
selection;
choosing a resource category and a sub-category to add a file to the
project;
adding information to the file, the information having the format
"<keyName> = <value>"; and
utilizing the file to generate a class in a child file of the file.

13-14. (Cancelled)

15. (Currently Amended) A computer-implemented method for reducing coding errors prior to runtime in the context of a managed code execution environment, comprising:

providing a developer with access to a plurality of managed code resources in the
managed code execution environment, the managed code execution
environment including a design program, a managed code infrastructure,
and a resource manager, the design program having tools for designing,
building, testing, and deploying applications, the managed code
infrastructure supporting cross-language compliant code, the design
program cooperating with the managed code infrastructure in production
of code that is compliant with a common language specification, the
common language specification comprising a set of basic language
features that are common across multiple applications, the set of basic
features being exposed in application programming interfaces to other
code, the resource manager managing resource information for languages
supported by the managed code infrastructure, the resource information
including information pertaining to resources that are compliant with the
common language specification; and

verifying that a resource identifier input by the developer corresponds to one of
the plurality of managed code resources by:

providing the developer with a collection of resource identifiers that include at least two identifiers that each identify a different language version of what is essentially the same resource; and receiving said resource identifier input from the developer in the form of a selection from the collection of resource identifiers.

16-25. (Cancelled)

26. (Currently Amended) The method of claim 15, wherein verifying comprises utilizing a computer processor that is a functional component of the computer to verify and wherein providing the developer with access to the plurality of managed code resources comprises the developer, during design time, interfacing with the design program to create code that incorporates a call to the resource manager and making, based at least in part on the call, a string resource request to the resource manager utilizing a "GetString (string KeyName)" call.

27. (Currently Amended) The method of claim 26, wherein providing the developer with a collection of resource identifiers comprises displaying the collection of resource identifiers in a pop-up window in a coding area and wherein the design program has an interface that allows the developer to adjust a status of the automatic delivery of resource information from an automatic setting, a manual setting, or a disabled setting, wherein for the automatic setting, a resource list automatically appears when an activator key is pressed, wherein for the manual setting, the developer selectively activates display of resource information by positioning a cursor at a location associated with information availability.

28. (Currently Amended) The method of claim 27, further comprising:
automatically inserting a string that corresponds to the selection into a programming code in the coding area; and performing a build-time check on the programming code, the build-time check determining whether resources have been addressed in error, the build-

time check comprising comparing resource identifier values in the programming code to a collection of valid values, wherein upon one of the resource identifier values not matching one of the collection of valid values, an error is submitted to the developer for correction.

29. (Previously Presented) A computer-implemented method for reducing coding errors prior to runtime in the context of a managed code execution environment, comprising:
- receiving from a developer an indication of a desired managed code resource;
 - communicating a resource request to a resource manager, the request including an indication of a key name and a string, the key name and the string both associated with the desired managed code resource;
 - displaying a collection of resource identifiers, each resource identifier corresponding to the key name and the string;
 - receiving from the developer a selection that corresponds to one of the resource identifiers in the collection of resource identifiers; and
 - automatically inserting the one of the resource identifier into a programming code that is in a coding area.
30. (Previously Presented) The method of claim 29, wherein the indication of a desired managed code resource comprises a call created in the programming code.
31. (Previously Presented) The method of claim 30, wherein the call comprises an entry of a resource object followed by an activator key.
32. (Previously Presented) The method of claim 31, wherein the activator key is a period.
33. (Previously Presented) The method of claim 31, wherein the activator key is a space bar.
34. (Previously Presented) The method of claim 31, wherein the activator key is a left parenthesis.

35. (Previously Presented) The method of claim 31, wherein displaying a collection of resource identifiers comprises displaying the collection in a pop-up window in the coding area.

36. (Previously Presented) The method of claim 35, wherein the pop-up window is located proximate to the call.

37. (Previously Presented) The method of claim 36, wherein stopping on one of the collection of resource identifiers in the pop-up window for a predetermined amount of time causes additional information to appear in a second pop-up window.